I hereby certify that this correspondence is being electronically transmitted on the date noted below to:

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

April 21, 2008
Date of Deposit

Amir N. Penn (Reg. No.: 40,767)
Name of applicant, assignee or
Registered Representative

Amir N Penn
Signature

April 21, 2008
Date of Signature

Our Case No. 9683/183

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Naoki Naruse et al.

Serial No. 10/814,662

Filing Date: March 31, 2004

For COMMUNICATION DEVICE AND PROGRAM

Examiner: Poltorak, Piotr

Group Art Unit No.: 2134

Confirmation No.: 5965

RESPONSE TO FINAL OFFICE ACTION DATED FEBRUARY 20, 2008

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants file the enclosed response to the final Office Action dated February 20, 2008 as follows:

Amendments to the Claims are reflected in the listing of the claims that begin on page 2 of this paper.

Remarks begin on page 8 of this paper.

AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application. Listing of the Claims:

1. (Previously Presented) A communication device for verifying whether application data is valid in order to determine whether to execute the application data on the communication device, the verification of the application data using a first file, a second file, and a third file, the first file comprising the application data, the second file comprising application validity data used to verify validity of the application data in the first file, and the third file comprising second file validity data calculated with a one-way function and used to verify the second file, the communication device comprising:

a receiver for receiving the first file, second file, and third file;

at least one processor for:

generating a second file calculated value using the one-way function, at least a part of the second file being input to the one-way function to generate the second file calculated value;

comparing the second file calculated value with the second file validity data in the third file; and

determining whether the second file is valid based on the comparing of the second file calculated value with the second file validity data in the third file; and

executing the application data on the communication device if the application data is verified using the application validity data in the second file and if it is determined that the second file is valid.

2. (Canceled)

3. (Canceled)

4. (Canceled)

5. (Canceled)

6. (Canceled)

7. (Previously Presented) The communication device of claim 1, wherein, if it is determined that the second file is valid, the processor further verifies whether the application data in the first file is valid using the application validity data in the second file.

8. (Previously Presented) The communication device of claim 7, where the application validity data is calculated on the basis of a one-way application validity function, the one-way application validity function using at least a part of the application data in the first file to generate the application validity data.

9. (Previously Presented) The communication device of claim 8, where the one-way application validity function comprises a hash function;

where the application validity data comprises a hash value;

where at least a part of the first file is input to the hash function to generate a calculated hash value; and

where using the application validity data in the second file to verify the validity of the application data in the first file comprises comparing the calculated hash value with the hash value in the second file.

10. (Previously Presented) The communication device of claim 9, where the one-way application validity function and the one-way function each comprise the same hash function.

11. (Previously Presented) The communication device of claim 1, where the application data comprises an application program.

12. (Previously Presented) The communication device of claim 1, where at least a part of the second file comprises independent data, the independent data being independent of the application data contained in the first file; and

where generating a second file calculated value using the one-way function comprises inputting the independent data to the one-way function to generate the second file calculated value.

13. (Previously Presented) The communication device of claim 12, where the independent data comprises certificate data for certifying authenticity of the application data, the certificate data being provided by a certificate authority.

14. (Previously Presented) The communication device of claim 13, where the third file is received from a trusted server;

where the second file is received from a content provider server, the content provider server being different from the trusted server; and

where the third file is used to verify the certificate data in the second file.

15. (Previously Presented) The communication device of claim 1, where the one-way function comprises a hash function;

where the second file validity data comprises a hash value;

where at least a part of the second file is input to the hash function to generate a calculated hash value; and

where comparing the second file calculated value with the second file validity data in the third file comprises comparing the calculated hash value with the hash value in the third file.

16. (Previously Presented) The communication device of claim 1, where the receiver receives the second and third files prior to receiving the first file; and

where the receiver receives the first file only after the at least one processor determines whether the second file is valid.

17. (Previously Presented) The communication device of claim 1, where the second file contains data indicating a location where a program, which is contained in the first file and run in the communication device, is stored.

18. (Currently Amended) A method for a communication device to verify whether application data is valid in order to determine whether to execute the application data on the communication device, the verification of the application data using [[the]] a first file, a second file, and a third file, the first file comprising the application data, the second file comprising application validity data used to verify validity of the application data in the first file, and the third file comprising second file validity data calculated with a one-way function and used to verify the second file, the method comprising:

receiving the first file, the second file and the third file at the communication device,

generating a second file calculated value using the one-way function, at least a part of the second file being input to the one-way function to generate the second file calculated value;

comparing the second file calculated value with the second file validity data in the third file;

determining whether the second file is valid based on the comparing of the second file calculated value with the second file validity data in the third file; and

executing the application data on the communication device if the application data is verified using the application validity data in the second file and if it is determined that the second file is valid.

19. (Previously Presented) The method of claim 18, further comprising, if it is determined that the second file is valid, verifying whether the application data in the first file is valid using the application validity data in the second file.

20. (Previously Presented) The method of claim 19, where the application validity data is calculated on the basis of a one-way application validity function, the one-way application validity function using at least a part of the application data in the first file to generate the application validity data.

21. (Previously Presented) The method of claim 20, where the one-way application validity function comprises a hash function;

where the application validity data comprises a hash value;

where at least a part of the first file is input to the hash function to generate a calculated hash value; and

where using the application validity data in the second file to verify the validity of the application data in the first file comprises comparing the calculated hash value with the hash value in the second file.

22. (Previously Presented) The method of claim 21, where the one-way application validity function and the one-way function each comprise the same hash function.

23. (Previously Presented) The method of claim 18, where the application data comprises an application program.

24. (Previously Presented) The method of claim 18, where at least a part of the second file comprises independent data, the independent data being independent of the application data contained in the first file; and

where generating a second file calculated value using the one-way function comprises inputting the independent data to the one-way function to generate the second file calculated value.

25. (Previously Presented) The method of claim 24, where the independent data comprises certificate data for certifying authenticity of the application data, the certificate data being provided by a certificate authority.

26. (Previously Presented) The method of claim 25, where the third file is received from a trusted server;

where the second file is received from a content provider server, the content provider server being different from the trusted server; and

where the third file is used to verify the certificate data in the second file.

27. (Previously Presented) The method of claim 18, where the one-way function comprises a hash function;

where the second file validity data comprises a hash value;

where at least a part of the second file is input to the hash function to generate a calculated hash value; and

where comparing the second file calculated value with the second file validity data in the third file comprises comparing the calculated hash value with the hash value in the third file.

28. (Previously Presented) The method of claim 18, where the second and third files are received prior to receiving the first file; and

where the first file is received only after determining whether the second file is valid.

29. (Previously Presented) The method of claim 18, where the second file contains data indicating a location where a program, which is contained in the first file and run in the communication device, is stored.

Remarks

1. Introduction

Claims 1 and 7-29 are pending.

2. Objections

The Office Action objected to claim 18 because "The first file" in claim 18 lacked antecedent basis. Applicants amend claim 18 to overcome the objection.

3. Rejections under 35 U.S.C. §103

Claims 1, 7-11, 15, 17-23, 27, and 29 were rejected under 35 U.S.C. §103(a) as obvious over Stallings (William Stallings, "Cryptography and Network Security," 2nd edition, 1998 ISBN: 0138690170) in view of Angelo (U.S. Patent Application No. 2003/0061487). Claims 1, 7-13, 15, 18-25, and 27 were rejected under 35 U.S.C. §103(a) as obvious over Angelo in view of Feghhi (Feghhi et al., "Digital Certificates Applied Internet Security," 1999, ISBN: 0201309807). Claims 14, 16, 26, and 28 were rejected under 35 U.S.C. §103(a) as obvious over Angelo in view of Feghhi and further in view of Fukumoto (U.S. Patent Application No. 2002/0073072).

The Office Action rejects independent claims 1 and 18 based on: (1) Stallings and Angelo; and (2) Angelo and Feghhi. Applicants respectfully contend that either combination fails to teach the invention as presently claimed.

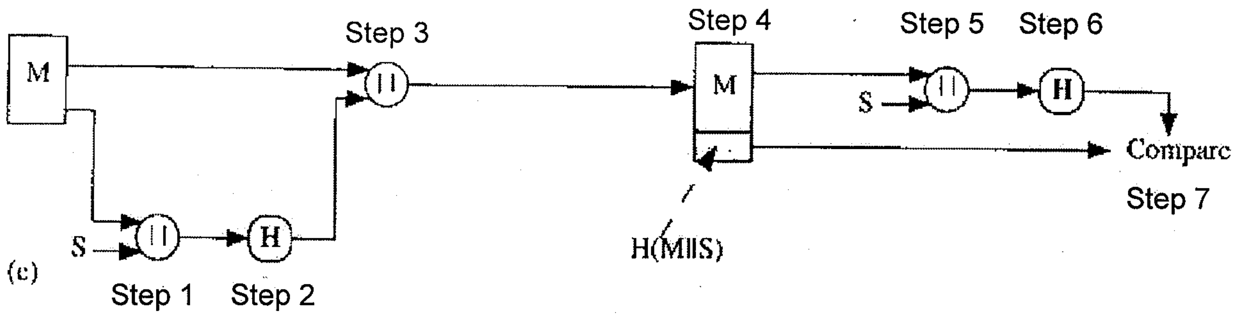
Rejection of the claims based on the Stallings and Angelo references

In rejecting claims 1 and 18 based on the Stallings and Angelo combination, the Office Action states the following:

> Stallings discloses a first file (message M) and a third file (hash of the first file: H(MIIS) that is put together in a second file, illustrated as a two color rectangle in Fig. 8.5 (e), for example.
>
> As per claims 1, 7, 11, 18-19, 23 the examiner interprets the first file to read on application data, bits value in the second file to read on application validity data and bits value in the third file as a second validity data.

Thus, in the rejection, the Office Action states that the first file is the message M, the third file is the hash of the Message M and the value S (H(MIIS)), and the second file is the combination of the first and third files.

Applicants respectfully contend that the Stallings reference fails to teach the three files as presently recited. For reference, Applicants reproduce Fig. 8.5(e) (with added text) and relevant portion describing Fig. 8.5(e) of the Stallings text:



e. This technique uses a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value S. A computes the hash value over the concatenation of M and S and appends the resulting hash value to M. Because B possesses S, it can recompute the hash value to verify. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

As described in the Stallings reference: M is the "message" (arguably the application file subject for ultimate verification); S is the "common secret value" (shared by transmitter "A" and receiver "B"); and H is the hash value. As shown in Fig. 8.5(e), the following steps occur:

step 1: combine M and S (identified as M||S);

step 2: create a hash value of the combination of M and S (identified as H(M||S);

step 3: append the hash value (identified as H(M||S) to M;

step 4: receive the combination of the hash value and M;

step 5: combine the received M with S (identified as M||S);

step 6: create a hash value of the combination created from step 5;

step 7: compare the hash value created in step 6 with the hash value received.

As shown from the above steps, the only difference from the teaching in the Stallings reference and the traditional prior art (such as disclosed in the Angelo reference) is the addition of the "common secret value" S prior to the hash step. However, this addition of S prior to the hash step does not result in a third file as presently claimed.

By way of background, this application relates to three files used in order to determine whether to execute an application file. The first file is the application file, the second file is used to validate the first file (such as a hash value for the first file), and the third file is used to verify the second file (using a one-way hash function). See claim 1 ("the first file comprising the application data, the second file comprising application validity data used to verify validity of the application data in the first file, and the third file comprising second file validity data calculated with a one-way function and used to verify the second file"); see also claim 18.

Given the three files as presently recited, Applicants cannot agree with the interpretation of the Stallings reference in the Office Action. First, Applicants disagree that the Stallings reference teaches the second file merely by combining the message M (interpreted as the first file) and the hash of the Message M and the value S (interpreted as the third file). Rather, the Stallings reference is more appropriately interpreted as only teaching two files, a first file (that includes the message M) and a second file (the hash of the Message M and the value S). Second, Applicants disagree that the second file (as properly interpreted) is "input to the one-way function to generate the second file calculated value". In its reasoning, the Office Action states that because the second file includes message M and because a hash value is generated at the destination using message M, the second file is input to the one-way function to generate the second file calculated value. Instead, as properly interpreted, the message M is the first file so that first file (instead of the second file) is input to the one-way function, similar to the teaching of the Angelo reference (described below). Third, nothing in the Stallings reference teaches verification of the hash of the Message M and the value S (H(MIIS)). As discussed above, the Stallings reference teaches, at best, that the second file is the hash of the Message M and the value S (H(MIIS)). However, the Stallings reference does not teach or even suggest any verification of this hash value. Instead, the entire focus is on verifying the Message M. Thus, as properly interpreted, the Stallings reference does not teach a third file as presently claimed and also does not teach "determining whether the second file is valid".

Moreover, the Angelo reference (used in combination with the Stallings reference) does not remedy the failings of the Stallings reference. As acknowledged in the Office Action, the Angelo reference does not teach or even suggest a third file or teach "determining whether the second file is valid." Specifically, the Office Action states that "Angelo does not disclose a third file comprising second file validity data calculated with a one-way function and used to verify the second file." Therefore, the combination of the Stallings and the Angelo reference does not render claims 1 and 18 obvious.

Rejection of the claims based on the Angelo and Feghhi references

In rejecting claims 1 and 18 based on the Angelo and Feghhi combination, the Office Action acknowledges the failings of the Angelo references as described above. The Office Action then reasons as follows:

> However, an ordinary artisan would readily recognize a security weakness of Angelo invention. Specifically, even though the process disclosed by Angelo verifies data application data authenticity it does not verify the source of the authentic data. The reason would have been clear to an ordinary artisan: verification of the application data in the first file depends on security of a public key that corresponds to a private key used in creating at the second file (the key signing the hash). A potential attacker could replace a legitimate file signing with its own private key and supplying the corresponding public key, and the verification process would also prove that the data (first file) is authentic. Of course, in this case the authentic data would be the data supplied by the attacker.

Applicants find several errors in the above-reasoning. First, where is the motivation that "an ordinary artisan would readily recognize a security weakness of Angelo invention (such as the security weakness of the "second file")"? In other words, if it is so "readily recognizable," where is the reference that teaches this? Instead, Applicants contend that the reasoning is mere hindsight. Second, even if someone had recognized the "security weakness of Angelo invention," those weaknesses are not remedied by combining the Angelo reference with the Feghhi reference.

The Feghhi reference generally teaches digital certificates. The Office Action generally cites the entire third chapter of the Feghhi reference in support of its conclusion that the Feghhi reference remedies the "security weakness of Angelo invention." Specifically, the Office Action states that because the Feghhi reference teaches digital certificates that it interprets as the third file, reasoning the following:

> Note that digital certificates are issued by Certification Authorities which are trusted organization (e.g. Fegghi, pg. 79-80). Thus, it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to provide a third file comprising application validity data given the benefit of increased security.

Applicants respectfully disagree. A digital certificate is used to verify a public key (used for encryption). The public key certificate (or identity certificate) is an electronic document which incorporates a digital signature to bind together a public key with an identity - information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to a particular individual. For example, "Alice" may have a public key for various transactions. The digital certificate can be used to verify that the public key associated with it is for "Alice".

Applicants frankly fail to see the relevance of the Feghhi reference. The Feghhi reference is directed to digital certificates, which are used to verify keys (and thereby validate the users of the keys). However, the invention as presently claimed is directed to verifying whether application data is valid in order to determine whether to execute the application data on the communication device. Given the lack of support for the assertions in the Office Action, Applicants contend that the rejection of claims 1 and 18 based on the combination of the Angelo and Feghhi references should be withdrawn.

Finally, the Office Action wholly fails to appreciate that the value that is analyzed in the second file (and used to compare with the third file) may be "independent data" (independent of the application data contained in the first file) (see claims 12 and 24), such as "certificate data for certifying authenticity of the application data, the certificate data being provided by a certificate authority." See claims 13 and 25. In this way, even if the content of the application data changes, the contents of the third file does not need to be updated since the data in the second file used for verification is independent of the application data. For example, the digital signature of the certificate in the second file (which is stored in the third file) does not need to be updated even if the application data changes. This is in contrast to what may happen between the first and second file. Specifically, if the application data in the first file is changed (such as updating an application program in the first file), the second file needs to be likewise updated to reflect the change (*e.g.*, the digital signature in the second file needs to be updated).

4. Conclusion

The Examiner is invited to contact the undersigned attorneys for the Applicant via telephone if such communication would expedite this application.

Respectfully submitted,

Amir N Penn

Amir N. Penn
Registration No. 40,767
Attorney for Applicant

BRINKS HOFER GILSON & LIONE
P.O. BOX 10395
CHICAGO, ILLINOIS 60610
(312) 321-4200